

Líneas de productos de software. Perspectivas de investigación en la Pontificia Universidad Javeriana de Cali

Nombre1 Apellido1*, Nombre2 Apellido2*, Nombre3 Apellido3*

*Departamento de Electrónica y Ciencias de la Computación,
Pontificia Universidad Javeriana, Cali, Colombia

Resumen—La mayoría de las industrias de software desarrollan productos sobre uno o pocos dominios en particular. Generalmente, los productos de un dominio comparten un amplio conjunto de características comunes, a las que deben ser adicionadas o modificadas funcionalidades. Las líneas de productos de software son una alternativa para gestionar de manera eficiente el desarrollo de familias de productos, reutilizando los elementos comunes y administrando la variabilidad. Sin embargo el desarrollo de una línea de productos puede ser complejo, por ejemplo en la definición de los modelos, la selección de configuraciones de los productos y el proceso de ensamblaje. Por ello se requieren metodologías y herramientas que hagan posible el desarrollo efectivo de líneas de productos de software. Es de nuestro interés aportar en la investigación y el desarrollo de las líneas de productos de software desde dos líneas de trabajo, cuyos retos y perspectivas se describen en este artículo.

Keywords—*Líneas de productos de software, defectos, modelos de variabilidad, configuración, ensamblaje*

I. INTRODUCCIÓN

En la industria de software, la mayoría de las empresas desarrollan productos para satisfacer necesidades en unos pocos dominios de aplicación. Por consiguiente, generalmente los productos de una empresa tienen muchas características y elementos comunes. Entonces, las prácticas de reutilización de componentes pueden acortar los tiempos de desarrollo de estos productos y mejorar la competitividad de las empresas. Sin embargo, para que la reutilización sea efectiva se requieren metodologías que, apoyadas en la tecnología, ayuden a administrar los procesos de reutilización durante el ciclo de vida del desarrollo del producto.

Las Líneas de Productos de Software (LPS) son un paradigma de producción que permite desarrollar y administrar, de manera eficiente, productos que tienen elementos comunes y variables. Así, a través de la reutilización de elementos las LPS ofrecen beneficios como menores tiempos de desarrollo, menores costos, mayor calidad y la flexibilidad para ofrecer a los usuarios productos que se ajusten a sus necesidades.

Son varias las etapas que incluyen las metodologías de desarrollo de líneas de productos de software. Entre ellas: La definición de modelos de variabilidad que especifican, para una familia de productos, las características comunes y variables, sus relaciones y las restricciones sobre las mismas. El diseño y desarrollo de los artefactos que soportan la línea de productos. La selección de elementos que satisfacen un conjunto particular de requerimientos, llamada configuración.

Y, finalmente, el ensamblaje de los elementos para obtener un producto en particular.

De otra parte, el desarrollo de líneas de productos puede ser complejo en varios sentidos, por ejemplo: Los modelos de variabilidad pueden incluir un gran número de características y restricciones, por ello son susceptibles de incluir errores semánticos. La cantidad de opciones de configuración puede ser tan grande que no sea posible hacer una lista que incluya todas las posibilidades, por lo tanto se hace necesario guiar las decisiones que se toman durante el proceso de configuración. Y el ensamblaje puede incluir elementos implementados con múltiples herramientas y bajo diversos paradigmas, lo cual hace necesario que el proceso de desarrollo siga una estrategia de ensamblaje de productos. Entonces, desarrollar metodologías y herramientas que den solución a estas y otras dificultades son temas importantes de investigación en el área de líneas de producto de software.

El Grupo de Investigación DESTINO de la Pontificia Universidad Javeriana está incursionando en el área de Líneas de Producto de Software a través de dos líneas de trabajo que se enfocan en aportar soluciones a las dificultades antes mencionadas. Nuestro objetivo es contribuir al desarrollo del paradigma LPS, de manera que pueda ser efectiva y ampliamente adoptado en la industria del software, en particular a nivel nacional y regional. Este artículo presenta los retos y perspectivas planteados para los proyectos que se están desarrollando en dichas líneas.

II. EXPERIENCIAS RELACIONADAS CON LÍNEAS DE PRODUCTOS DE SOFTWARE

Las líneas de productos de software (LPS) son del interés del equipo de profesores que pertenecen al grupo de investigación Destino, por ser este un tema en que se combinan la investigación teórica con la posibilidad de aplicar los conceptos desde un punto de vista práctico en la industria de desarrollo de software de la región y el país.

En particular, el interés está enfocado en dos líneas de trabajo: la calidad de los modelos de variabilidad y la adopción del paradigma de líneas de productos de software con casos de aplicación reales.

A continuación se detallan ambas líneas de trabajo.

II-A. Calidad de los modelos de variabilidad. ¿Cómo identificar de una manera genérica los defectos, sus causas y correcciones en los modelos de variabilidad?

Las líneas de productos de software se representan por medio de modelos de variabilidad, pues éstos sirven como un punto de comunicación intermedia entre los usuarios finales y los desarrolladores de la LPS [1]. Estos modelos permiten identificar por ejemplo, cuáles características son comunes a todos los productos de la línea de productos, cuáles características son variables; cuáles características deben ir juntas en todos los productos y cuáles características no pueden ir al mismo tiempo en un producto [1].

Los modelos de variabilidad se diseñan en una de las primeras etapas del desarrollo de la línea de productos y son un insumo importante para el resto del proceso de la ingeniería de línea de productos. Sin embargo, a medida que aumenta la complejidad de los modelos de variabilidad, es posible que quienes los elaboran introduzcan en ellos sin intención defectos semánticos. Los defectos semánticos son imperfecciones que afectan la capacidad del modelo de variabilidad para expresar de manera correcta el dominio que representa la línea de productos [2].

Rincón *et al.* [3], [4] proponen, para modelos de variabilidad representados con la notación de los modelos de características (en inglés *Feature Models* [1]), un método que identifica defectos semánticos y sugiere correcciones. Para ello, toman de la programación por restricciones el concepto de MCSes (*Minimal Correction Subsets*) y lo aplican para identificar las correcciones de los defectos semánticos encontrados. Específicamente, los MCSes corresponden a las dependencias que deben ser eliminadas de los modelos de variabilidad para solucionar al menos un defecto.

Antes de entregar los resultados al usuario final, el método transforma a lenguaje natural las correcciones identificadas para cada defecto. De esta manera se obtienen los defectos semánticos de los modelos de variabilidad y las correcciones para cada defecto en un lenguaje fácilmente entendible por quienes diseñan los modelos.

Emplear MCSes para identificar correcciones de defectos semánticos en modelos de características es importante cuando ésta es la notación empleada para modelar las líneas de productos. Sin embargo, los modelos de variabilidad pueden ser expresados en otras notaciones como OVM [5], o Dopler [6], que no son consideradas en el trabajo de Rincón *et al.* [3], [4]. Así mismo, es necesario evaluar el método propuesto considerando aspectos como la satisfacción del usuario final, la utilidad de la solución, la escalabilidad, entre otros. La línea de trabajo orientada hacia la calidad de los modelos de variabilidad busca considerar estos aspectos con el fin de generalizar el método propuesto por Rincón *et al.* [3], [4] para la identificación de defectos, causas y sus correcciones.

II-B. Adopción del paradigma de líneas de productos de software

Con el fin de aplicar alguna de las metodologías propuesta para la adopción de líneas de productos de software en un dominio local, iniciamos un proyecto de investigación que nos permitirá, para un primer año, explorar y formular una

línea de productos de software, donde se evaluarán diferentes dominios para seleccionar aquel sobre el cual se desarrollará la LPS. Se definirán los requerimientos y el modelo conceptual de la LPS, y el modelo de variabilidad que representa las relaciones de sus elementos así como las restricciones entre ellos. Se aplicarán métodos de verificación a los modelos de variabilidad para garantizar que éstos se encuentren libres de defectos semánticos. Finalmente, se expresarán los modelos de variabilidad en un lenguaje formal, que sirvan como entrada a un configurador.

A partir de éste proyecto, desarrollaremos un trabajo más ambicioso, que permita en tres años *Elaborar un conjunto de prácticas, artefactos y criterios genéricos (framework) que apoye la configuración guiada de productos y el ensamblaje de componentes en líneas de productos de software*. Estos aspectos, la configuración y el ensamblaje de productos de software, son fundamentales en nuestros intereses de investigación.

1) *Configuración de productos*: En la ingeniería de líneas de productos el desarrollo de una línea de productos se divide en dos procesos: la **ingeniería de dominio** y la **ingeniería de aplicación**. En la ingeniería de dominio se analiza el área de aplicación en la que se desea crear la línea de productos, se define su alcance, se identifican los elementos comunes y variables que contendrá la línea de productos, y se diseñan y construyen los *core assets*, que soportan la línea de productos [5]. Por su parte, en la ingeniería de aplicación, los usuarios seleccionan características de la línea de productos a partir de las cuales se configuran y ensamblan productos que pertenecen a la línea de productos [5]. La idea básica de las líneas de productos es desarrollar *core assets* durante la ingeniería del dominio que luego permitan obtener productos individuales durante la ingeniería de aplicación [7].

Una **configuración** es una selección de elementos que deben satisfacer tanto los requerimientos del usuario como las restricciones del dominio de la línea de productos [8]. Por esta razón, configurar un producto de una línea de productos debe ser un proceso guiado paso a paso que debe permitir satisfacer a la vez las necesidades o ambiciones del usuario que realiza la configuración y las restricciones propias del dominio de la línea de productos [2]. Además, el proceso de configuración debe satisfacer algunas características no funcionales como: tiempo de respuesta, independencia del orden de selección, adaptación al entorno cambiante, selección de características óptimas de acuerdo a los intereses del usuario, entre otras [9]. Si en una configuración ninguno de sus elementos genera conflictos con otros elementos de la línea de productos, entonces la configuración es válida y representa los requerimientos de un producto que pertenece a la línea de productos, mientras que en caso contrario la configuración es inválida.

En este sentido, nuestra investigación busca definir técnicas de apoyo a la configuración y establecer la forma de recibir los requerimientos de entrada de los usuarios, la información histórica y las restricciones relacionadas con el modelo de variabilidad. De igual forma, se desarrollará un sistema que permita a los usuarios configurar productos de la LPS y se definirán prácticas y criterios que deben seguirse en la línea de productos para facilitar la configuración guiada de los productos de software.

2) *Ensamblaje de componentes*: Una vez se configuran los productos de la línea de productos, es necesario establecer mecanismos para construir el producto de software configurado [10]. Este proceso, conocido como **ensamblaje**, consiste en tomar core assets de manera que minimizando el esfuerzo necesario para obtener cada producto en particular, sea posible obtener productos finales de la línea de productos [11].

De esta forma, en nuestro proyecto de investigación, definiremos criterios y prácticas para la elaboración de artefactos reutilizables que puedan ser ensamblados de manera automática o semiautomática para generar productos que hagan parte de una LPS. Adicionalmente, se propondrán criterios para la definición de la arquitectura de líneas de productos de software de referencia y para la selección de las tecnologías con la que se implementarán los componentes reutilizables. Por último, se definirán prácticas y criterios para el ensamblaje de los componentes de software reutilizables y se implementarán estrategias para relacionar las elecciones hechas por el usuario en la configuración con los componentes reutilizables, de modo que éstos se puedan ensamblar con el fin de obtener el producto configurado.

III. CONCLUSIONES

Sección final que presenta un resumen del artículo y sus resultados concretos, analizados en una perspectiva más amplia, por ejemplo en cómo estos resultados pueden ser aplicados a otras aplicaciones o sectores.

También podría incluir caminos de extensión o trabajo futuro.

AGRADECIMIENTOS

Sección opcional para agradecimientos, por ejemplo fuentes de financiación, colaboración con empresas, universidades, etc.

REFERENCIAS

- [1] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and S. P. Peterson, "Feasibility Study Feature-Oriented Domain Analysis (FODA). Technical Report," Software Engineering Institute, Carnegie Mellon University, Tech. Rep., 1990.
- [2] C. Salinesi and R. Mazo, "Defects in Product Line Models and how to identify them," in *Software Product Line - Advanced Topic*, A. Elfaki, Ed. InTech, 2012, ch. 5, pp. 97–122.
- [3] L. Rincón, G. Giraldo, R. Mazo, C. Salinesi, and D. Díaz, "Método para Identificar Correcciones de Defectos en Modelos de Características que representan Líneas de Productos," in *Proceedings of the 40th Latin American Computing Conference*, Montevideo, 2014.
- [4] L. F. Rincón Perez, G. L. Giraldo Gómez, R. Mazo, C. Salinesi, and D. Díaz, "Subconjuntos Mínimos de Corrección para explicar características muertas en Modelos de Líneas de Productos. El caso de los Modelos de Características," in *Proceedings of the 8th Colombian Computer Conference (CCC)*, Armenia-Colombia, 2013.
- [5] K. Pohl, G. Böckle, and F. J. van Der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., 2005.
- [6] D. Dhungana, R. Rabiser, P. Grünbacher, K. Lehner, and C. Federspiel, "DOPLER: An Adaptable Tool Suite for Product Line Engineering," in *Proceedings of the 11th International Software Product Line Conference*, ser. SPLC'07, Kyoto-Japan, 2007, pp. 151–152.
- [7] R. Rabiser, P. OLeary, and I. Richardson, "Key activities for product derivation in software product lines," *Journal of Systems and Software*, vol. 84, no. 2, pp. 285–300, Feb. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121210002700>

- [8] J. White, D. Benavides, D. Schmidt, P. Trinidad, B. Dougherty, and A. Ruiz-Cortés, "Automated diagnosis of feature model configurations," *Journal of Systems and Software*, vol. 83, no. 7, pp. 1094–1107, Jul. 2010.
- [9] R. Mazo, C. Dumitrescu, C. Salinesi, and D. Díaz, "Recommendation Heuristics for Improving Product Line Configuration Processes," in *Recommendation Systems in Software Engineering*, 2014, pp. 511–537.
- [10] S. Deelstra, M. Sinnema, and J. Bosch, "Product derivation in software product families: a case study," *Journal of Systems and Software*, vol. 74, no. 2, pp. 173–194, Jan. 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121203003121>
- [11] P. Clements and L. M. Northrop, *Software Product Lines: Practices and Patterns*, 1st ed. Addison-Wesley Professional, 2001.